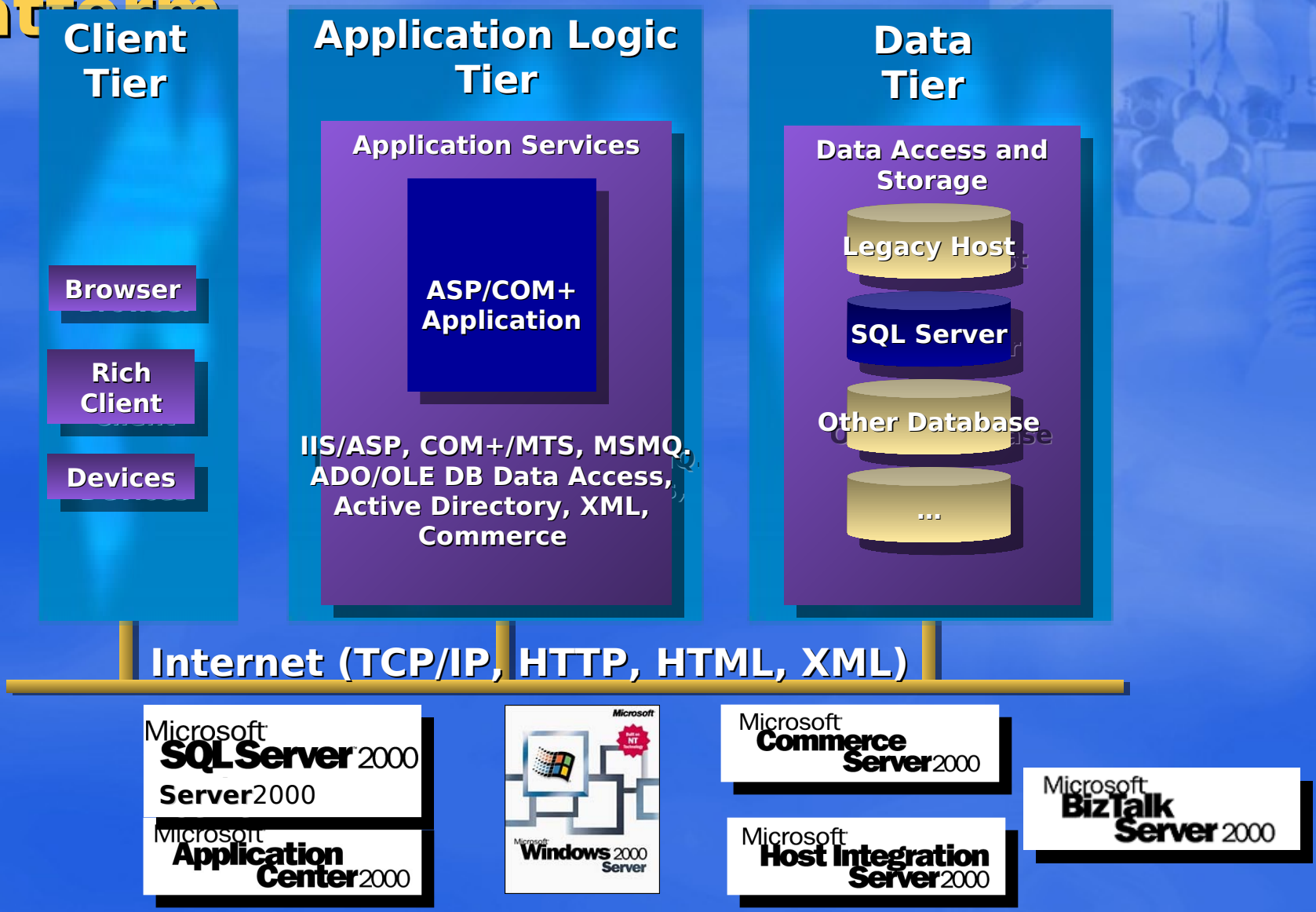
Microsoft And XML

Matt Domo
Group Program Manager
Data Access and XML
Microsoft Corporation


POWER UP
Windows DNA 2000
Readiness Conference
featuring SQL Server 2000

Windows DNA 2000

Next Generation Web Application Platform



Agenda

- 50,000 Feet – Why XML is Important
- Why/What we hear from Customers
- Architecture
- Loosely Coupled
- Strategy
- Microsoft’s Policy
- Required Steps

50,000 Feet

- XML will be the format and the data model for all information flowing across applications, via messages, and through the web
- Microsoft is betting on this

Why Is XML Important?

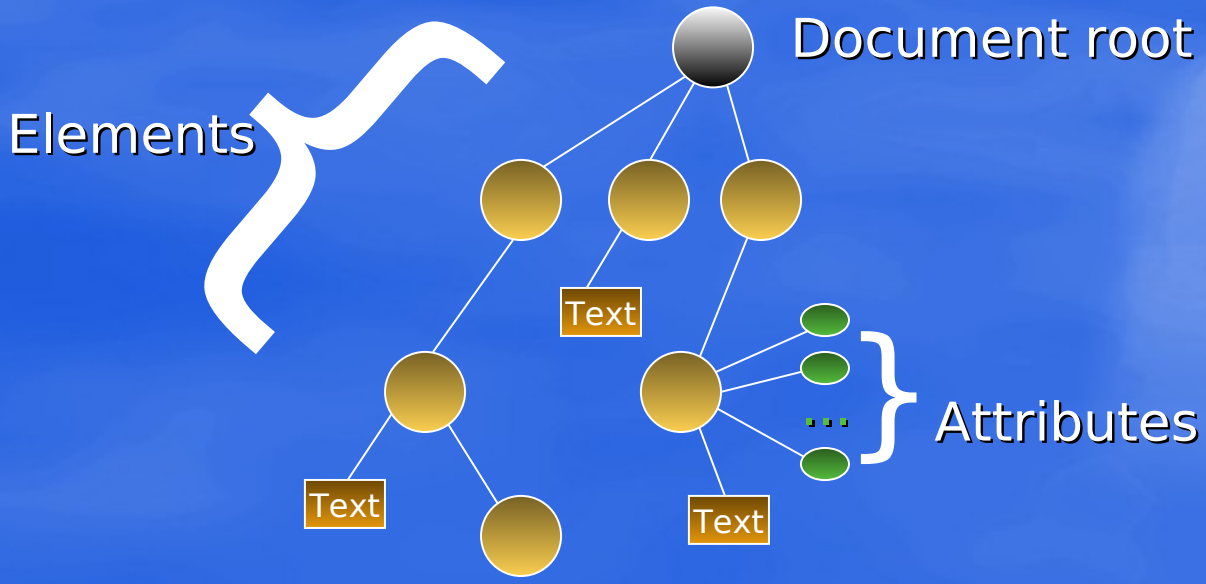
- XML is simple
- XML continues the Web's open standards
- XML makes data more important than code
- XML enables powerful Web applications

XML Supports Many Data Models

- XML as a hierarchy
- XML as a row set
- XML as complex data

XML As A Hierarchy

- DOM
- XSLT for declarative transforms



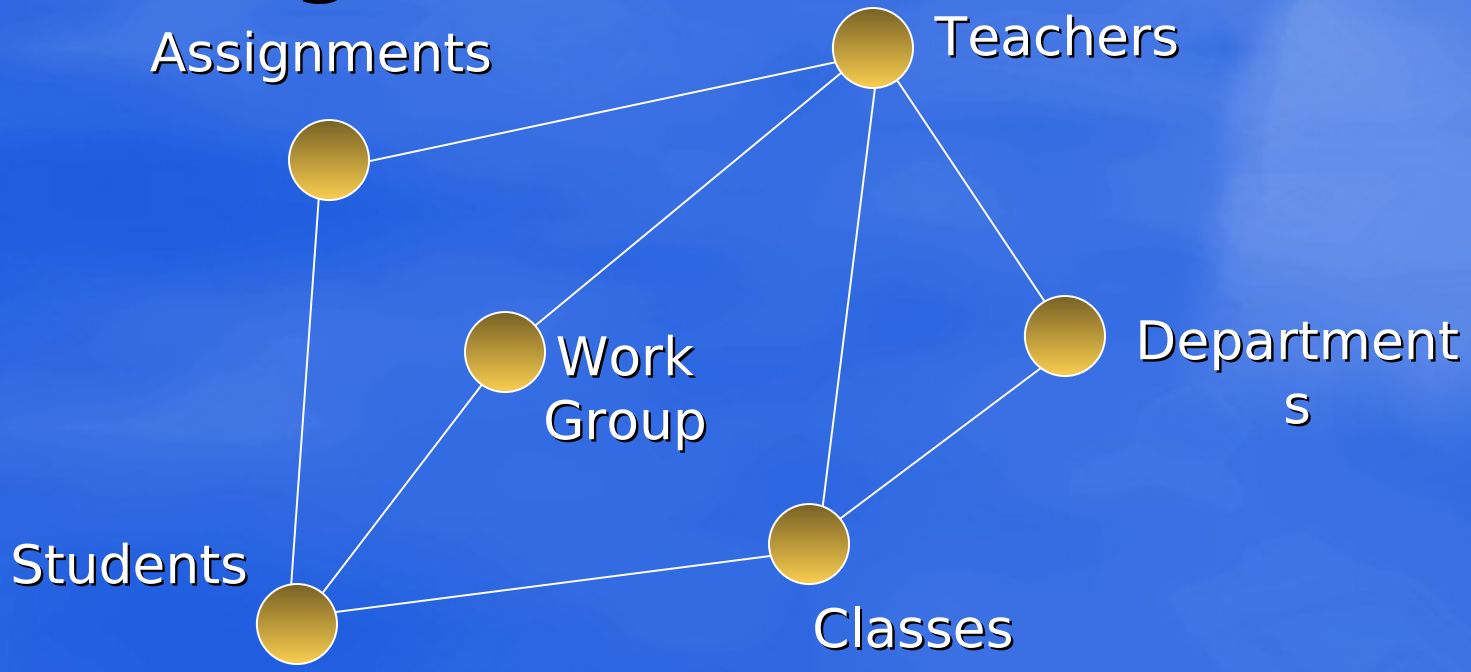
XML As A Row Set

- Well understood
- Used for “remoting” data
  - Data Binding

	A	B	C	D
1				
2				
3				

XML As Complex Data

- **More complex data relationships**
  - **Often relational data mapped to single object**
- **A “logical view” of the data**



Common Themes

- Usually an intermediate representation of data
- Not constrained by the physical storage format
- Not tied to any programming language or environment

Next: XML As A Message

- Used as a transport layer
- Transient data - doesn’t exist in any permanent data store
- Independent of implementation

What We Hear From Customers

- Why the Web works at all
  - Existing standards are easy, flexible, platform neutral, ubiquitously available
  - Rich enough for most user interface
  - Links are discoverable
- It is Scalable
  - The coarse grained nature of the interactions
  - Minimal need for maintaining client state
  - Caching at every level of the system

What We Hear From Customers

- It *must be* Loosely Coupled
  - Formats and Protocols drive agreements
  - Interactions cannot know about the target's implementation which *will* vary over
  - **Space**: Buying books, tracking stocks
  - **Time**: Customer's address fields, subclasses
  - Remember, you cannot change client code when the server's implementation changes and all implementations change

What We Hear From Customers

- We need a similar architecture for data
  - Easy, Flexible, Platform Neutral, Ubiquitous
  - Rich enough for most information
  - Discoverable
- Scalable
  - Coarse Grained, Cached, Asynchronous
- Loosely Coupled
  - Neutral to the recipients

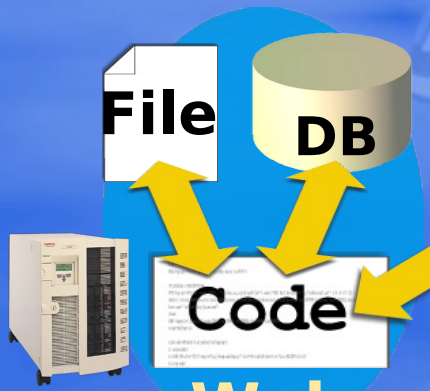
Loosely Coupled Today?

- Today's model is tightly coupled
  - Tightly coupled to an Application
    - Specific to each site's "methods"
    - Stored Procedures, HTTP GET with Parameters
  - *Or* Tightly coupled queries to a database
    - SQL which largely exposes the data model of the implementation. Views are a palliative, not a solution. For example, consider using Views for NEWSML
  - Neither model supports discovery
    - This is a productivity barrier which is leading to ugly unreliable technologies

First And Second Generation Web Apps

Web Server

1-1 correspondence



In-house systems

Web Server

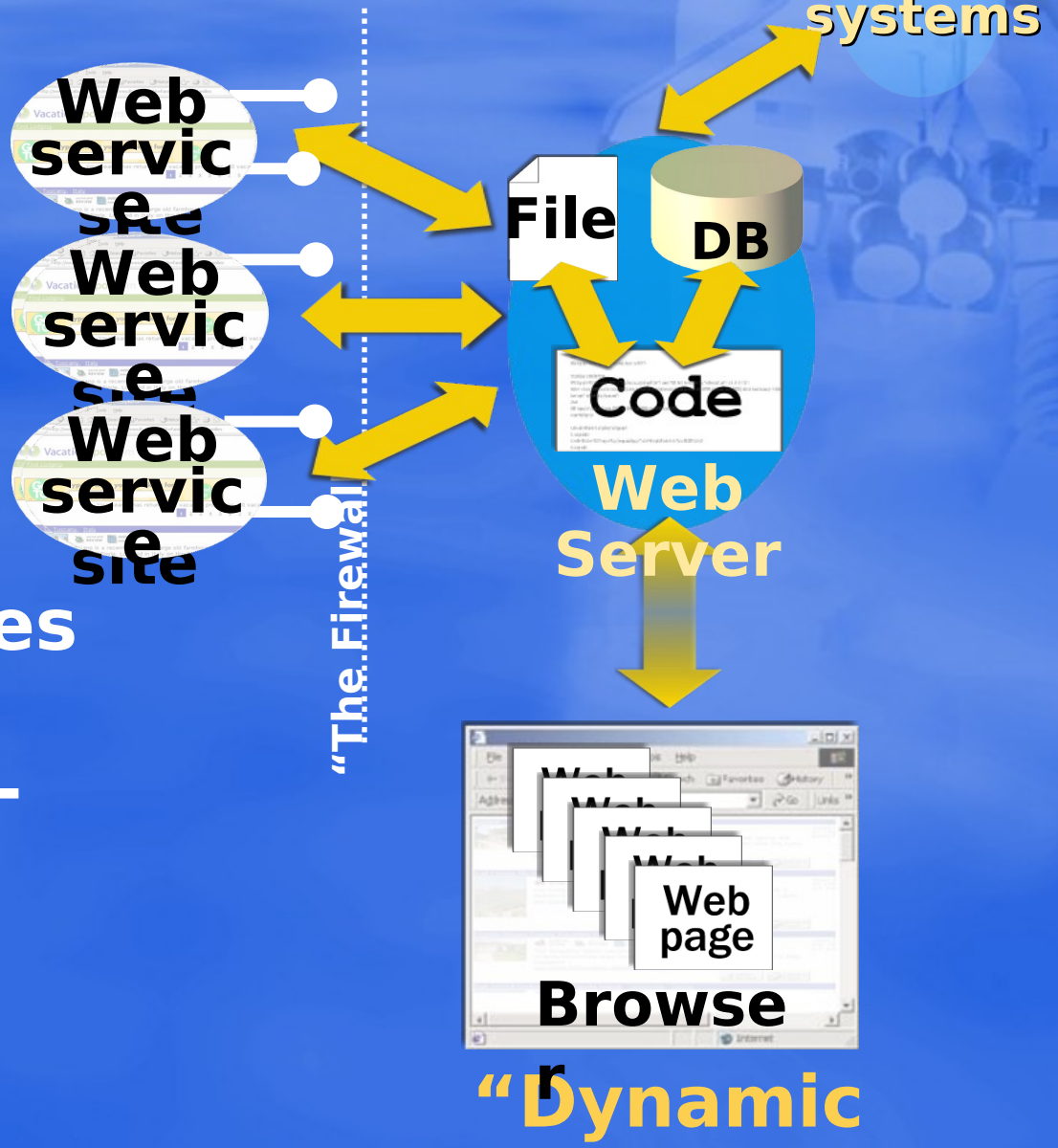
“Dynamic Pages”

Third Generation Web

Accessing another site today = HTML “screen scraping” or “your architect calls my architect”

Tomorrow, external Web sites become building blocks using XML

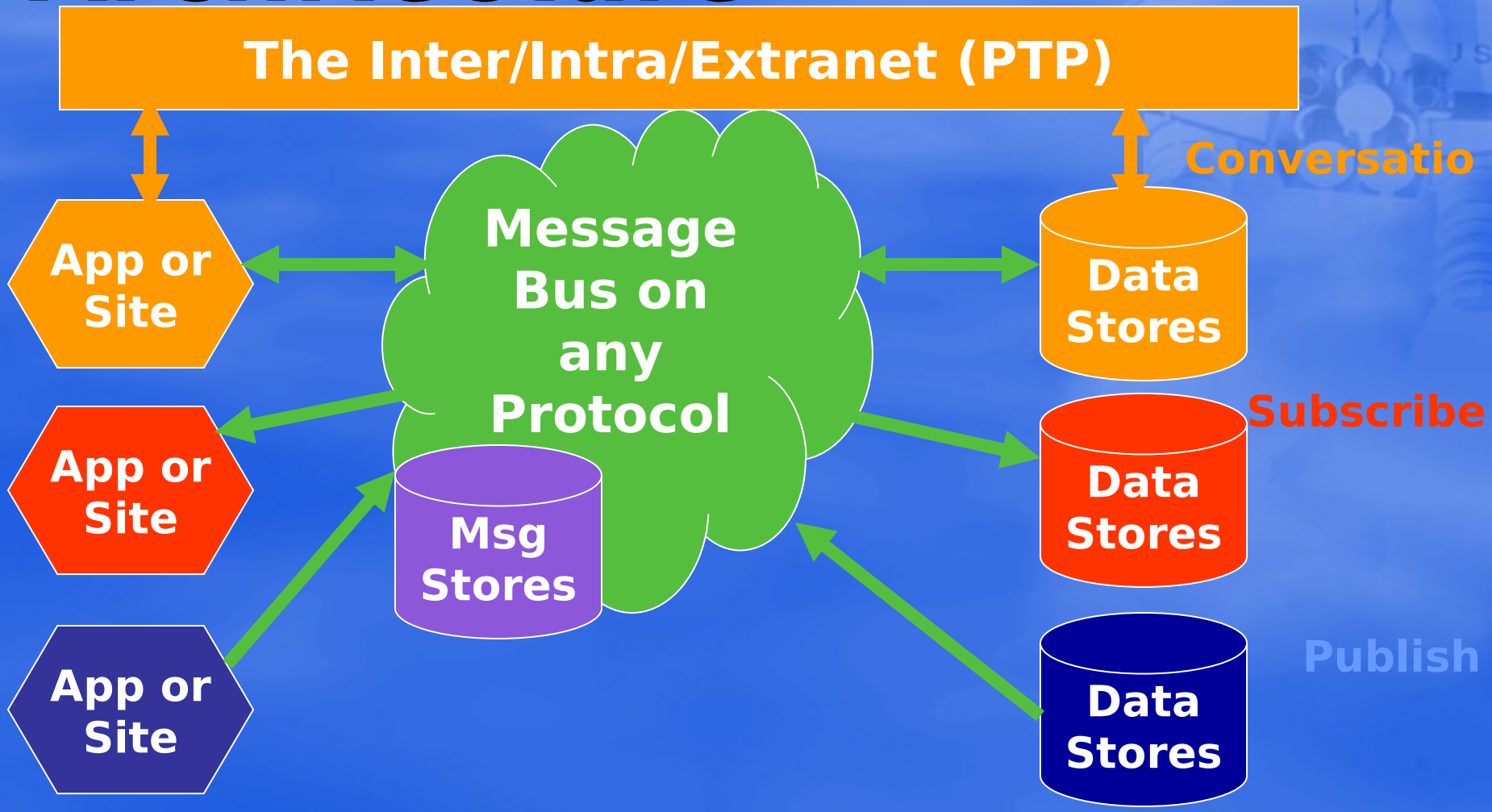
- Web Services
- Megaservices



Architecture

- Two basic ways to move information between applications and either applications or Data stores
  - Point to Point
    - Request Response
    - Methods
    - Still can allow for asynchronous fetching and intelligent caching
    - Requires a discovery mechanism
  - Message Based
    - Publish and/or Subscribe
    - Event Architecture

Forthcoming Architecture



Examples Of Web Services

Location Services	Maps, routing, nearby locations…
Shopping Services	Order tracking, supply chain, auctions, coupons…
Information Services	Headlines, weather, horoscopes, TV times…
Communication Services	Email, instant messages…

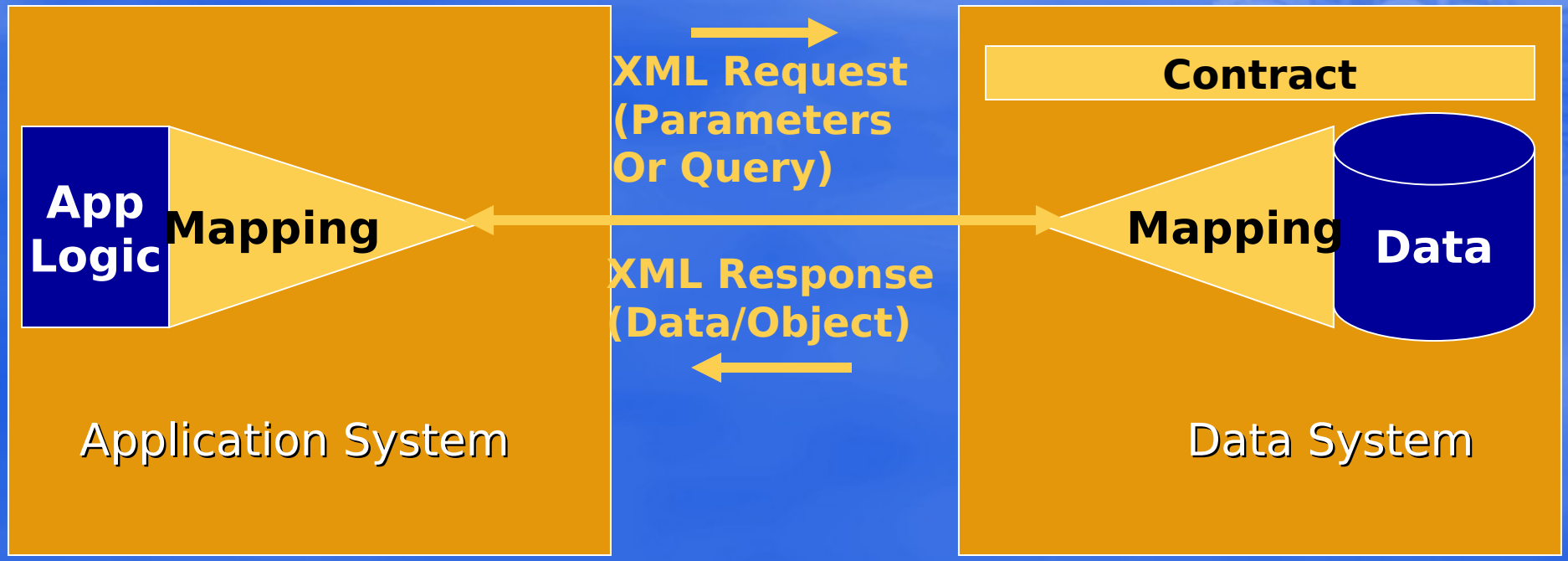
Data Stores

- Data stores can play several roles:
  - PTP communication. In essence the data store is one or more “application” objects
  - Provider of multiple “XML Views”, data models that can be queried or updated and that map to the XML the recipient expects to receive
  - Filtered “subscribers” to published data flowing from the message bus
  - Publishers of data to the message bus

Loosely Coupled Data Stores

- How to query these Data Stores
  - View the data as a set of rich XML views which can describe entire networks of information
  - Enable the client to query and update these views
  - Discover which views are available and queriable
  - Enable the client to request a query as a single XML message
  - Allow the client not to know the difference between an implementation using a data store and one using a simple file

Architecture Drill In



Discussion Of Drill In

- **Contract**
  - **Describes the set of interactions you can have with this application, data provider, or site. Some will be simple request response exchanges of XML messages Some will be more complex sequences**
- **Mapping**
  - **Enables the implementation to map from the message as proscribed by the contract to the actual implementation. XSLT will be widely used for this**

Microsoft's Policy

- Support an Open Standards based architecture for communication between applications, sites, and data (W3C for formats, IETF for Protocols)
  - XML (Messages, Data)
  - SOAP (Invocation, Self describing messages)
  - XSLT (Mapping, Limited Queries)
  - XML Schema (Contracts, Message Metadata)
  - XPath, XML Query Language (Queries)
- Ship tracking as best we can
- Rev and provide migration if and

Microsoft's Policy

- Encourage the participation of all key systems vendors such as IBM and Oracle in the standards process
- Work closely with the W3C to build the common standards we all need for this architecture
- All Web, All the time!
  - Innovate and release technology preview components on the web
  - Track the latest W3C standards
  - Updated frequently so visit our site often

Microsoft's Policy

- Provide the best tools for building and interacting with this architecture
- Provide the best components and services for interacting with this architecture
- Provide the best data stores for interacting with this architecture
- Provide a message bus that enables the routing and mapping of messages across any components
- Build all formats as XML formats

Microsoft's Policy

- BizTalk Framework
  - Guidelines for building Schemas
  - Grammar for defining message routing
- BizTalk.Org
  - Place to learn about the Framework
  - Place to publish and find Schemas
  - Community for exchanging ideas
- BizTalk Server
  - Server for routing and mapping messages

Required Steps

- Schema
  - Messages in XML must be able to describe data and objects
  - Requires a richer schema language than DTDs
  - The work is underway, but it must be finished
  - We'll provide migration from XML-DATA reduced and DTDs
- Queries
  - An XML Query language is required
  - Work is just beginning on this front
  - We'll use XPath and XSLT until we can

Required Steps

- PTP Invocation
  - We plan to support and use SOAP for this
- Discovery and Conversations
  - IETF efforts are required
  - Until then we'll support a XML based solution
  - When the IETF builds a standard we'll support it and provide migration tools for it

Conclusion

- XML will be the format and the data model for all information flowing across applications, via messages, and through the web
- Last year this was still in doubt
- It no longer is

Resources

- Microsoft XML Developer Center
  - http://msdn.microsoft.com/xml/
- BizTalk
  - http://www.biztalk.org
  - http://www.microsoft.com/industry/biztalk/
- World Wide Web Consortium
  - http://www.w3c.org/xml/

Questions

POWER UP

Microsoft